

A Prior Information Enhanced Extraction Framework for Document-level Financial Event Extraction

Haitao Wang, Tong Zhu, Mingtao Wang, Guoliang Zhang & Wenliang Chen[†]

School of Computer Science and Technology, Soochow University, Suzhou 215006, China

Keywords: Event extraction; Information extraction; Financial event; Event detection; Event argument extraction

Citation: Wang, H.T., et al.: A prior information enhanced extraction framework for document-level financial event extraction. Data Intelligence 3(3), 460-476 (2021). doi: 10.1162/dint_a_00103

Received: February 11, 2021; Revised: April 8, 2021; Accepted: May 18, 2021

ABSTRACT

Document-level financial event extraction (DFEE) is the task of detecting events and extracting the corresponding event arguments in financial documents, which plays an important role in information extraction in the financial domain. This task is challenging as the financial documents are generally long text and event arguments of one event may be scattered in different sentences. To address this issue, we proposed a novel **Prior Information Enhanced Extraction** framework (PIEE) for DFEE, leveraging prior information from both event types and pre-trained language models. Specifically, PIEE consists of three components: event detection, event argument extraction, and event table filling. In event detection, we identify the event type. Then, the event type is explicitly used for event argument extraction. Meanwhile, the implicit information within language models also provides considerable cues for event arguments localization. Finally, all the event arguments are filled in an event table by a set of predefined heuristic rules. To demonstrate the effectiveness of our proposed framework, we participated in the share task of CCKS2020 Task 4-2: Document-level Event Arguments Extraction. On both Leaderboard A and Leaderboard B, PIEE took the first place and significantly outperformed the other systems.

1. INTRODUCTION

Event Extraction (EE) aims to identify different types of events and their corresponding arguments in text. In the financial domain, EE provides valuable structured information for investment analysis and asset management. To promote financial event extraction, the 14th China Conference on Knowledge Graph and

[†] Corresponding author: Wenliang Chen (Email: wlchen@suda.edu.cn; ORCID: 0000-0002-5429-6084).

Semantic Computing (CCKS2020) set Task 4-2^① for document-level financial event extraction (DFEE). The organizer collected documents from financial news and announcements, and required the participants to identify the event types and extract event arguments from the documents.

In recent years, event extraction has attracted increasing attention due to its vast application and significant efforts have been devoted to it. However, most existing studies merely extract arguments within the sentence scope [1, 2, 3], dubbed as sentence-level EE (SEE). For document-level EE, these methods provide sub-optimal solutions because the event arguments are often scattered across different sentences in a document and global information should be exploited to enhance the model. As shown in Figure 1, most of the text data contain more than 500 Chinese characters. Under this circumstance, independently processing each sentence in the document destroys the integrity of events. Therefore, a document-level EE framework is vital to extract events from such long documents.

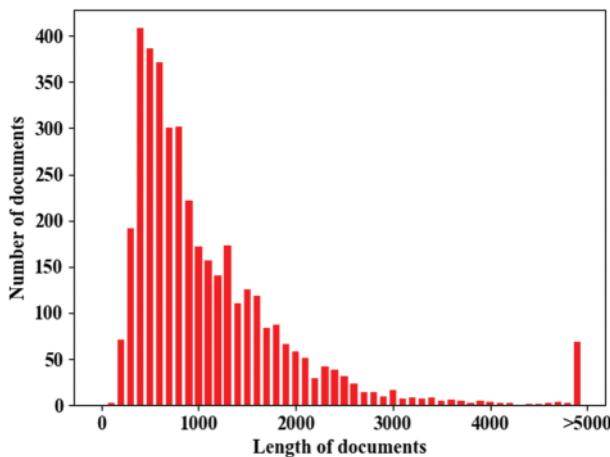


Figure 1. The text length distribution of data in CCKS2020 Task 4-2.

In this paper, we proposed a Prior Information Enhanced Extraction framework (PIEE) for document-level financial event extraction, which can be decomposed into three steps: event detection, event argument extraction, and event table filling. Specifically, in event detection we first identified the event type of the document. Then, we utilized the event type as prior information for sentence-level event argument extraction. In this paper, we explored three paradigms for event argument extraction. With prior type information, all the three paradigms obtained consistent performance improvement. Moreover, inspired by the recent success of pre-trained language model (PLM) which is trained on large corpus and provides implicit prior information, we explored different language models for event argument extraction. Finally, event table filling integrated all event arguments extracted from different sentences by a set of heuristic rules.

^① https://www.biendata.xyz/competition/ccks_2020_4_2/

In summary, our contributions are summarized as follows:

- We proposed a novel prior information enhanced extraction framework (PIEE) for document-level financial event extraction, which is comprised of three steps: event detection, event argument extraction and event table filling.
- We utilized event type as explicit prior information for sentence-level event argument extraction. Meanwhile, we explored the implicit prior information in different language models for event argument extraction.
- In CCKS2020 Task 4-2, our system achieved 0.83007 *F1*-score on Leaderboard A and 0.66996 *F1*-score on Leaderboard B, both ranking the first place.

2. RELATED WORK

Event extraction has achieved great progress in recent years. However, most research [4, 5, 6] focused on sentence-level event extraction (SEE), and document-level event extraction (DEE) was less concerned. Yang et al. [7] and Zheng et al. [8] proposed two different frameworks for DEE. The former method (DCFEE) extracts event arguments in the form of SEE and combines the results of SEE into DEE by a key event detection and arguments-completion strategy, which depends on event triggers. The latter one establishes an end-to-end framework Doc2EDAG based on multiple transformer models and exploits an entity-based directed acyclic graph to implement the DEE without any elaborately designed rules. But at the same time, Doc2EDAG also faces problems such as complex structure, low efficiency, and large resource occupation.

In the stage of event argument extraction, both of them regard it as a sequence labeling problem similar to NER, where BiLSTM-CRF [9] is a classic model to address this issue. Beyond that, with the successful application of machine reading comprehension (MRC) in many NLP problems [10, 11], MRC is also used in NER tasks with the advantage of significant prior information of the entity category. Recently, Yu et al. [12] applied the Biaffine model to NER tasks and achieved the state-of-the-art performance on eight corpora.

In addition, compared to GloVe [13] and ELMo [14], recent language model BERT can capture more contextual and semantic information from texts. To mitigate the drawbacks of masking strategies in BERT, BERT-wwm [15] uses the Whole Word Masking (WWM) and ERNIE [16] designs the entity-level strategy and the phrase-level strategy to integrate external knowledge. RoBERTa [17] further proposes the dynamic masking strategy and removes the next sentence prediction task. Relative positional encoding is also employed in NEZHA [18] to enhance the encoding ability.

Inspired by the above work, we proposed a prior information enhanced extraction framework for document-level financial event extraction. In contrast to DCFEE and Doc2EDAG, we first discovered events in texts, which helps identify the event arguments in the subsequent stages. To improve the performance of event argument extraction, advanced technologies in NER and recent language models were also introduced in our model. Furthermore, from the view of structure, our framework is simpler and faster. And the event triggers are not necessary in PIEE.

3. DATA

This section presents data analysis and describes how to preprocess data.

3.1 Data Analysis

In order to have a comprehensive understanding of the data in the shared task, we listed statistical information. Figure 2 presents the co-occurrence distribution of different event types in the training data, including Bankruptcy Liquidation (BL), Equity Freeze (EF), Equity Underweight (EU), Equity Overweight (EO), Equity Pledge (EP), Asset Loss (AL), Accident (AC), Leader Death (LD), and External Indemnity (EI). We can conclude that all the events in one document share the same event type. This observation greatly simplifies the process of event type identification.

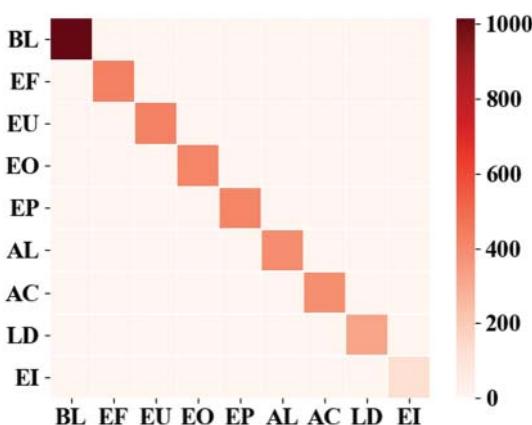


Figure 2. Co-occurrence distribution of event types in training data.

Figure 3 further shows the distribution histogram of the number of documents and instances in each event type. It can be observed that the event types are divided into two categories: one is that the event occurs only once in the document like Bankruptcy Liquidation, and the other is that the event can occur more than once in the same document such as Equity Pledge. This fact also contributes to subsequent event table filling.

In summary, we can draw the following two conclusions:

- Each document contains only one type of event.
- There is only one event in the document which describes BL, AL, AC, LD and EI, and documents introducing EU, EO, EF and EP usually contain more than one event.

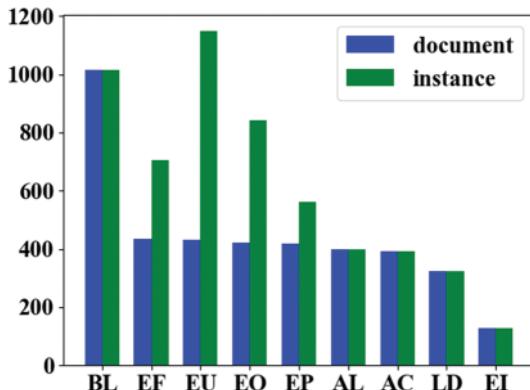


Figure 3. Number of documents and instances in each event type.

3.2 Data Preprocessing

The data of this evaluation task mainly come from financial announcements and news on the Internet. Inevitably, there are noises in the crawled texts. Thus, it is necessary to clean the data for better system construction.

As shown in Table 1, the original data contain the escape symbols and tags of HTML, which hinder the system's semantic understanding of texts. We restore them except `
`, which is specially replaced with a single space considering that `\n` is a special flag when splitting the document.

Table 1. Escape symbols and tags of HTML in the evaluation data.

 	"	&apos	&	>	<	
\s	"	'	&	>	<	\n

Moreover, in order to minimize the length of the text as possible, the continuous repeated punctuation, extra spaces and Web links are removed. We also converted traditional texts into simplified texts, and converted punctuation from SBC case to DBC case to construct more standardized data. Finally, all documents are divided into multiple sentences with a maximum length of 500 Chinese characters and event arguments in the sentence are tagged with BIO (Begin, Inside, Other) scheme in the training data.

4. METHODOLOGY

In this section, we introduce the details in our proposed framework. First of all, we needed to detect which event types are described in the documents. Then, we treated event argument extraction as a sequence labeling problem. At last, some heuristic strategies were applied to fill in the event tables.

4.1 Event Detection

In the research of distantly supervised relation extraction, Riedel et al. [19] assumed: If two entities have a relation, at least one sentence can express that relation in all sentences containing those two entities. Inspired by this classical assumption, we also assumed: If a document contains an event type, at least one sentence from this document can fully describe that event type.

In the previous research of event extraction, event trigger is often used to recognize the event type. However, no trigger words are explicitly provided in real scenarios. We assumed that in the document describing the event, there is at least one trigger word implicitly, and the sentence where the trigger word is located must be able to pick out the event type described in this document. Under this assumption, each document can be considered to be a sentence bag.

Figure 4 shows the architecture of event detection. Sentences from the same document $\{s_1, s_2, \dots, s_n\}$ are first transformed into distributed representations by looking up the pre-trained char embeddings. Then, sentence encoder such as CNN and LSTM is applied to extract deep semantic features $\{h_1, h_2, \dots, h_n\}$ for text classification. Similar to the research in relation extraction, sentences from the same document are regarded as one bag, and there are three strategies to represent a document d : ONE (at least one sentence), ATT (selective attention over sentences), and MAX (cross-sentence max pooling).

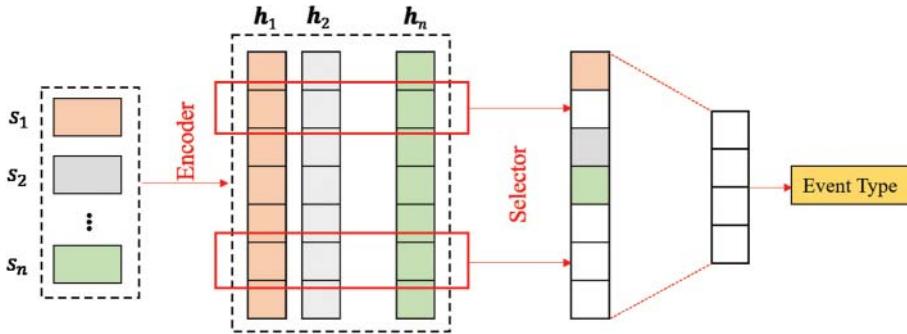


Figure 4. The architecture of event detection.

4.1.1 ONE

Zeng et al. [20] selected the most valuable sentence to represent the whole sentence bag d and the highest probability sentence is defined as follows:

$$\begin{aligned} \mathbf{o}_i &= \mathbf{W}\mathbf{h}_i + \mathbf{b} \\ j^* &= \operatorname{argmax}_i \frac{\exp(\mathbf{o}_i)}{\sum_k \exp(\mathbf{o}_k)} \\ \mathbf{d} &= \mathbf{h}_{j^*} \end{aligned} \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{n_e \times h_i}$, n_e is the number of event types and \mathbf{h}_i is the size of hidden units.

4.1.2 ATT

Following Lin et al. [21], to exploit the information of all available sentences, we can use the attention mechanism to aggregate sentence-level features. The score a_i measures how well the input sentence s_i and the target event type e matches can be obtained by the following equation:

$$a_i = \mathbf{h}_i \mathbf{W}_a \mathbf{r}_e \quad (2)$$

where \mathbf{W}_a is a weighted diagonal matrix, and \mathbf{r}_e is the representation of event type e .

Then, the representation of the document \mathbf{d} is computed as a weighted sum of sentence-level features:

$$\mathbf{d} = \sum_i \frac{\exp(a_i)}{\sum_k \exp(a_k)} \mathbf{h}_i \quad (3)$$

4.1.3 MAX

Jiang et al. [22] claimed that critical information can be also inferred implicitly from all sentences, so a max pooling operation is employed to capture the most valuable features in various aspects from all sentences. Formally, the document-level feature \mathbf{d} is computed as follows:

$$\mathbf{d} = \max(\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n) \quad (4)$$

Finally, event type is predicted by the representation of document \mathbf{d} and cross-entropy is used as the objective function to optimize the models.

4.2 Event Argument Extraction

For event argument extraction, many classic methods of sequence labeling task can be used to extract event arguments in texts. In order to make full use of prior information of event type, we concatenated sentences and the representation of the corresponding event type before encoding. Thus, all sentences from the same document share the same event type predicted by event detection. Based on such input representation, we proposed three PLM-based architectures for sentence-level event argument extraction: PLM-CRF, PLM-MRC, and PLM-Biaffine.

4.2.1 PLM-CRF

BiLSTM-CRF is a classic model to address the NER task and has once achieved the state-of-the-art result in accuracy. Since pre-trained language models like BERT can capture deeper semantic and contextual information, in our PLM-CRF, the input sequence of PLM consists of event type and sentence. With the help of multiple layers of transformers in PLM, sentence can make full interaction with prior information.

Given the output of PLM $\{r_1, r_2, \dots, r_m, x_1, x_2, \dots, x_l\}$, where r_i is the output of event type and x_i is the output of sentence, $X = \{x_1, x_2, \dots, x_l\}$ is then used as the input of the CRF layer. For a sequence of predictions $y = \{y_1, y_2, \dots, y_l\}$, we define its score as in Equation (5):

$$s(\mathbf{X}, y) = \sum_{i=0}^l \mathbf{A}_{y_i, y_{i+1}} + \sum_{i=0}^l (\mathbf{W}\mathbf{X})_{i, y_i}^T \quad (5)$$

where $\mathbf{A} \in \mathbb{R}^{(n_t+2) \times (n_t+2)}$ is a matrix of transition scores and $\mathbf{W} \in \mathbb{R}^{n_t \times h}$ is used to calculate the scores of each label for each token, n_t is the number of BIO tags and h is the hidden size of PLM.

During training, we maximized the log-probability of the correct tag sequence. In the testing stage, we used Viterbi algorithm to decode the sequence.

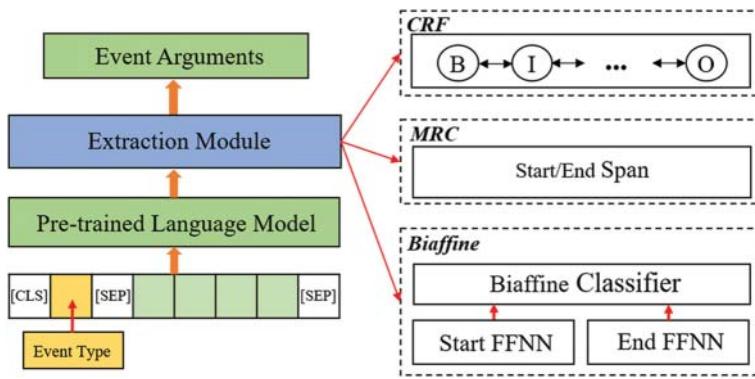


Figure 5. The architecture of event argument extraction.

4.2.2 PLM-MRC

At present, many NLP tasks can be converted into machine reading comprehension (MRC) problems, and inspired by Li et al. [23], we proposed a simplified version of MRC to address event argument extraction.

First of all, we manually constructed some queries for event roles in different event types. For example, for Pledgor in Equity Pledge, the corresponding query is “who is the pledgor in equity pledge”. Similar to the operation in PLM-CRF, we also concatenated the query and sentence before PLM encoding.

Then, given the representation of sentence $X = \{x_1, x_2, \dots, x_l\}$ output from the BERT, we can compute the probabilities of each token being a start index and an end index respectively as follows:

$$\begin{aligned} \mathbf{P}_s &= softmax(\mathbf{W}_s \mathbf{X} + \mathbf{b}_s) \\ \mathbf{P}_e &= softmax(\mathbf{W}_e \mathbf{X} + \mathbf{b}_e) \end{aligned} \quad (6)$$

where $\mathbf{W}_s \in \mathbb{R}^{h \times 2}$ and $\mathbf{W}_e \in \mathbb{R}^{h \times 2}$, h is the hidden size of PLM.

In the prediction stage, all valid combinations for a start index and an end index are regarded as the span of event arguments, where there are no other start/end indices between them.

4.2.3 PLM-Biaffine

The Biaffine model is widely used in dependency parsing [24] and Yu et al. [12] first applied this architecture to address the NER task. Following their work, we also used the Biaffine model to extract event arguments in texts.

Same as the operation in PLM-CRF, we first obtained the sentence representation $X = \{x_1, x_2, \dots, x_t\}$ from PLM. After that, two feedforward neural networks (FFNN) were used to generate the representations for the start/end of the spans. Then a Biaffine model was applied to predict possible event roles for each span, including a special role named as NA, which means that the current span is not a valid event argument. Specifically, the score of event role for span $\langle i, j \rangle$ was computed as follows:

$$\begin{aligned} \mathbf{h}_s^i &= \mathbf{W}_s \mathbf{x}_i + \mathbf{b}_s \\ \mathbf{h}_e^j &= \mathbf{W}_e \mathbf{x}_j + \mathbf{b}_e \\ s(i, j) &= \mathbf{h}_s^{i\top} \mathbf{U} \mathbf{h}_e^j + \mathbf{W}_h (\mathbf{h}_s^i \oplus \mathbf{h}_e^j) + \mathbf{b} \end{aligned} \quad (7)$$

where \mathbf{h}_s^i and \mathbf{h}_e^j are the start/end representation of token i and j , $s(i, j)$ is the score distribution for span $\langle i, j \rangle$ among nr event roles. $\mathbf{W}_s \in \mathbb{R}^{h \times d}$, $\mathbf{W}_e \in \mathbb{R}^{h \times d}$, $\mathbf{U} \in \mathbb{R}^{h \times n_r \times h}$, $\mathbf{W}_h \in \mathbb{R}^{2d \times n_r}$ are trainable parameters in the Biaffine model.

When decoding, the event role of each span is one of the highest scores and we ranked all non-NA spans by their category scores in a descending order. Entities in the sentence are regarded as event arguments only if their spans do not clash the boundaries of higher ranked entities, or there is no inclusive relation between higher ranked entities and them.

4.3 Event Table Filling

After obtaining the event types and event arguments in the document, we designed some heuristic strategies to convert the results of SEE to DEE. According to corollaries mentioned in Section 3.1, all event types can be divided into two categories: one type one event (OTOE) and one type multiple events (OTME).

In the training data, events in OTOE always appear in the plain texts. The combination of valid event arguments with minimum internal distance^② is selected as the event in document. Leader Death is a special event type in OTOE since it is obvious to find event triggers in the sentences, such as “去世”, “逝世”, and “辞世” (all mean pass away). The distance between triggers and event arguments is also considered while computing the internal distance.

^② We define the internal distance as the sum of distances between all event arguments.

In the OTME scenario, events mainly appear in the table. Thus, we first tended to use keywords, such as “本次增持股票数量(万股)” (number of overweight equity), to locate the table, and parse table content with the help of regular expressions and event arguments extracted by models. If no event is found by table parsing, events are generated by the same method in OTOE.

Additionally, there are some universal strategies. For example, we compared the longest common sequence (LCS) to determine whether a company name is a full name or an abbreviation. To reserve the special token (mostly *
*) in the final answer, we checked all answers which contain space and do not appear in the original text, and restored them to their original form.

5. EVALUATION

This section presents the experimental results on the evaluation data, and the detailed analysis. We compared different variants in event detection and event argument extraction mentioned in Section 4.

5.1 Data Set and Experimental Setup

Experiments are conducted on CCKS2020 Task 4-2 data set. This data set contains 9 event types. In the training data, there are 3,956 documents containing 5,521 events, which are annotated by distant supervision [25, 26]. Validation data and testing data are used for online evaluation on Leaderboard A and Leaderboard B, which contain 750 documents and 28,096 documents, respectively. In order to achieve better robustness and anti-noise capability, we used a 5-fold cross-validation to train each model.

In the experiments of event detection, we used Adam to optimize parameters with a learning rate of 0.001 and a minibatch size of 32. The hidden size of BiLSTM and CNN are both 256. While extracting event arguments, the learning rate is set to 2e-5 in PLM layers and 2e-4 in other layers. The maximum epoch of PLM-CRF, PLM-MRC and PLM-Biaffine is respectively 5, 3 and 5. In particular, the output size of FFNNs are both 256 in PLM-Biaffine.

5.2 Experimental Results of Event Argument Extraction

Table 2 shows the results of different models mentioned in Section 4.1. It is obvious that MAX-based models achieved the highest accuracy as MAX can capture the most valuable information from all sentences in the document. On the other hand, since predictive features could be diluted by noises in the document, ATT is not as good as MAX. Among three strategies, ONE shows the worst performance both in CNN-based models and BiLSTM-based models, which means that it is not enough to use the information of a single sentence to represent the full text in text classification. It is worth noting that the data of this evaluation task mainly come from financial announcements, which usually have a title that summarizes the full text.

Thus, a simplified solution is to exploit the information of the title to classify the document. Then we used the first sentence of each document for event detection. Compared to ONE, it works better, but not the best.

Table 2. Different models for event detection.

	CNN	BiLSTM	BERT
First-Sentence	0.98031	0.98158	0.98081
ONE	0.97524	0.94045	-
ATT	0.98233	0.97251	-
MAX	0.98560	0.98988	-

5.3 Experimental Results of Event Argument Extraction

For three paradigms of event argument extraction, we all used BERT-wwm-Chinese as pretrained language model. In order to exploit the global information, the results of event detection were regarded as prior information, which was shared by all sentences from one document. As shown in Table 3, it is obvious that models using prior information of event types always perform better, which shows global information of a document is beneficial to event extraction and it is necessary to detect event type before event arguments extraction.

Table 3. Different model variants for event argument extraction.

Models	F1-score	Training Time/Epoch
PLM-CRF †	0.82503	31min
PLM-CRF	0.84033	31min
PLM-MRC †	0.00000	63min
PLM-MRC	0.84777	63min
PLM-Biaffine †	0.82691	18min
PLM-Biaffine	0.84772	18min

Note: † means no prior event type information is utilized.

Among all models, although PLM-MRC yields the best performance, PLM-Biaffine still achieves similar results, and has enormous advantage of training speed. Thus, we selected PLM-Biaffine as the basic model and further explored different PLMs in order to make full use of implicitly prior information within PLMs. From Table 4, we can observe NEZHA-large performs best, which directly leads to the result that we used only the combination of NEZHA-large and PLM-Biaffine (NEZHA-Biaffine) in the final competition.

Table 4. Different PLMs for PLM-Biaffine.

PLM	F1-score
BERT-base	0.84615
BERT-wwm	0.84772
BERT-wwm-ext	0.84977
ERNIE	0.84298
RoBERTa-wwm-ext	0.85546
RoBERTa-wwm-ext-large	0.86533
NEZHA-large	0.86693

5.4 Online Results

According to the above experimental results, BiLSTM+MAX and NEZHA-Biaffine were selected as our final models. The detailed results are listed in Table 5, and it shows that our model (PIEE) is effective. Moreover, since the online result of Bankruptcy Liquidation, Asset Loss, Accident, Leader Death and External Indemnity are always 0 on the final testing data, we trained the new model on the data of rest event types again, which increased the results from 0.66247 to 0.66996.

Table 5. Top 5 teams on Leaderborad A and Leaderborad B.

Leaderborad A		Leaderborad B	
Teams	F1-score	Teams	F1-score
PIEE	0.83007	PIEE	0.66996
Rank 2	0.81411	Rank 2	0.65043
Rank 3	0.80578	Rank 3	0.63469
Rank 4	0.78422	Rank 4	0.61530
Rank 5	0.78359	Rank 5	0.60464

6. CONCLUSION AND FUTURE WORK

In this paper, we proposed a Prior Information Enhanced Extraction Framework (PIEE) for document-level financial event extraction, which consists of three components: event detection, event argument extraction and event table filling. In our solution, we show that it is necessary to detect event types first in DEE, which is helpful to extract event arguments as explicit prior information. Moreover, we explore the implicit prior information of different PLMs in event argument extraction. For Document-level Event Argument Extraction in CCKS2020 Task 4-2, our system achieved 0.83007 F1-score and 0.66996 F1-score on Leaderboard A and Leaderboard B, respectively, which are both the highest scores, showing the advantages of our framework.

Nevertheless, our framework could be further improved due to its potential limitations and deficiencies. On the whole, PIEE is a pipeline framework, which might cause error propagation and accumulation. For example, the performance of event argument extraction largely depends on the result of event detection. Moreover, it is inflexible to fill in the event tables using heuristic strategies. This is where we need further improvement in the future.

ACKNOWLEDGEMENTS

The research is supported by the National Natural Science Foundation of China (No. 61936010 and No. 61876115). This work was partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

chinaXiv:202211.00376v1

AUTHOR CONTRIBUTIONS

H.T. Wang (htwang2019@stu.suda.edu.cn) contributed to data set statistics, design of experiments and manuscript writing. T. Zhu (tzhu7@stu.suda.edu.cn) contributed to data set statistics, experiments with different pretrained language models and manuscript writing. M.T. Wang (wangmt@suda.edu.cn) contributed to strategies in extracting equity freeze event, table content parsing and data annotation. G.L. Zhang (glzhang@stu.suda.edu.cn) contributed to data set statistics, bad case analysis, data annotation and manuscript revision. W.L. Chen (wlchen@suda.edu.cn) contributed to data set statistics, design of the whole framework and manuscript writing. All authors have made meaningful and valuable contributions in revising and proofreading manuscripts.

DATA AVAILABILITY STATEMENT

The data sets generated and/or analyzed during the current study are not publicly available due to the fact that the data sets are produced by expert consultants of the Institute of Automation, Chinese Academy of Sciences and the Ant Financial Services Group based on their own experience. The publicly released version of the data sets needs the consent of all expert consultants, and they are available from the corresponding author on reasonable request.

REFERENCES

- [1] Chen, Y., et al.: Event extraction via dynamic multi-pooling convolutional neural networks. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pp. 167–176 (2015)
- [2] Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 300–309 (2016)

- [3] Nguyen, T.M., Nguyen, T.H.: One for all: Neural joint modeling of entities and events. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp. 6851–6858 (2019)
- [4] Liu, X., Luo, Z., Huang, H.Y.: Jointly multiple events extraction via attention-based graph information aggregation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 1247–1256 (2018)
- [5] Zhang, T., Ji, H.: Event extraction with generative adversarial imitation learning. arXiv preprint arXiv:1804.07881 (2018)
- [6] Lample, G., et al.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270 (2016)
- [7] Yang, H., et al.: Dcfee: A document-level Chinese financial event extraction system based on automatically labeled training data. In: Proceedings of ACL 2018, System Demonstrations, pp. 50–55 (2018)
- [8] Zheng, S., et al.: Doc2edag: An end-to-end document-level framework for Chinese financial event extraction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 337–346 (2019)
- [9] Lample, G., et al.: Neural architectures for named entity recognition. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 260–270 (2016)
- [10] Levy, O., et al.: Zero-shot relation extraction via reading comprehension. In: Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017), pp. 333–342 (2017)
- [11] Li, X., et al.: Entity-relation extraction as multi-turn question answering. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 1340–1350 (2019)
- [12] Yu, J., Bohnet, B., Poesio, M.: Named entity recognition as dependency parsing. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 6470–6476 (2020)
- [13] Pennington, J., Socher, R., Manning, C.D.: GloVe: Global vectors for word representation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 1532–1543 (2014)
- [14] Peters, M., et al.: Deep contextualized word representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 2227–2237 (2018)
- [15] Cui, Y., et al.: Pre-training with whole word masking for Chinese BERT. arXiv preprint arXiv:1906.08101 (2019)
- [16] Sun, Y., et al.: Ernie: Enhanced representation through knowledge integration. arXiv preprint arXiv:1904.09223 (2019)
- [17] Liu, Y., et al.: RoBERTa: A robustly optimized BERT pretraining approach. arXiv preprint arXiv:1907.11692 (2019)
- [18] Wei, J., et al.: Neural contextualized representation for Chinese language understanding. arXiv preprint arXiv:1909.00204 (2019)
- [19] Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, pp. 148–163 (2010)
- [20] Zeng, D., et al.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1753–1762 (2015)

[21] Lin, Y., et al.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 2124–2133 (2016)

[22] Jiang, X., et al.: Relation extraction with multi-instance multi-label convolutional neural networks. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 1471–1480 (2016)

[23] Li, X., et al.: A unified MRC framework for named entity recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5849–5859 (2020)

[24] Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)

[25] Mintz, M., et al.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2, pp. 1003–1011 (2009)

[26] Chen, Y., et al.: Automatically labeled data generation for large scale event extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 409–419 (2017)

AUTHOR BIOGRAPHY

Haitao Wang is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities, and document-level event extraction.

ORCID: 0000-0003-2531-149X



Tong Zhu is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities, and document-level event extraction.

ORCID: 0000-0002-5433-8504



Mingtao Wang is a graduate student at the School of Computer Science and Technology, Soochow University. He is a research assistant in the university now. He is interested in information extraction, dialogue system and knowledge base question answering.

ORCID: 0000-0002-8838-1519



Guoliang Zhang is a postgraduate student at the School of Computer Science and Technology, Soochow University. He is interested in multi-format information extraction, especially semantic relation extraction between entities or events, and event extraction.

ORCID: 0000-0002-3639-0712



Wenliang Chen received his Bachelor's degree in Mechanical Engineering and PhD degree in Computer Science from Northeastern University in 1999 and 2005, respectively. He joined Soochow University in 2013 and is currently a professor in the university. Prior to joining Soochow University, he was a research scientist at the Institute for Infocomm Research of Singapore from 2011 to 2013. From 2005 to 2010, he worked as an expert researcher in NICT, Japan. His current research interests include parsing, machine translation, and machine learning.

ORCID: 0000-0002-5429-6084